

Contingut

1. Exemple pràctic:	1
2. Exercici ABSTRACTE	2
3. Exercici DEMANATS	3
4. Exercici PRÉSTECES.....	4
5. EXERCICI ESCALA.....	5
6. EXERCICI SUCURSALS.....	8

Per a cadascun dels exercicis primer crea un SCHEMA.

1. EXEMPLE PRÀCTIC:

Quan posem una restricció de valor per defecte el valor ha de ser un literal, no podem assignar el valor d'una funció. No obstant això, podem assignar la data del sistema per defecte si triem per al camp el tipus adequat i utilitzem la constant `CURRENT_TIMESTAMP` (no és una funció: no porta parèntesi) com en l'exemple.

Normalment quan definim una columna només podem posar una restricció de columna en la seua definició (si té unes altres, les posaríem al final). El problema és que les restriccions de `NOT NULL` i `DEFAULT` només es poden definir a nivell de columna pel que es poden posar les dues juntes en la definició.

D'altra banda, `AUTO_INCREMENT` és una propietat que poden tindre els camps `INT` (compte: seria un camp indexat) i només es pot incloure en la definició de columna i pot anar juntament amb una altra restricció. Cal tindre en compte que una columna `AUTO_INCREMENT` quan li donem valor hem de posar-li `NULL` i el sistema ja li assigna el valor que corresponga; no hem d'introduir un altre valor (encara que es podria).

Prova el següent script (còpia i pega en un query del WorkBench):

```
create schema if not exists EXEMPLE;

use EXEMPLE;

create table if not exists TABLA1 (

    CAMPO1 int primary key auto_increment,

    FECHA1 timestamp NOT NULL default CURRENT_TIMESTAMP,
```

```
FECHA2 timestamp default CURRENT_TIMESTAMP) engine=innodb;

insert into TABLA1 values (null, null, null), (null, null, default),
(null, default, '1990-01-01'), (null, default, default);

select * from TABLA1;
```

Depèn de la versió de MySql pot ser que, encara que no tinga restricció de NOT NULL (cas de FECHA2), en assignar al camp el valor NULL, el sistema assigne el valor per defecte.

2. EXERCICI ABSTRACTE

Creeu les següents taules en l'ordre adequat. Trieu els dominis dels camps que siguin variats de manera que siguin coherents en totes les taules:

A(a1,a2,a3)
 CP:{a1,a2}
 CAj: {a2} \rightarrow C

B(b1, b2,b3,b4)
 CP:{b1,b2}
 VNN: {b3}

C(c1,c2,c3)
 CP:{c1}
 UNI:{c2,c3}
 CAj: {c1,c2} \rightarrow B

```
create schema abstracte;
use abstracte;

create table B (
b1 int,
b2 int,
b3 date not null,
b4 varchar(10),
constraint pk_B primary key(b1,b2)
) engine=innodb;

create table C (
c1 char(5) primary key,
c2 int,
c3 int,
constraint fk_C_B foreign key(c2,c3) references B(b1,b2),
constraint uni_c2c3 unique (c2,c3)
) engine=innodb;

create table A (
a1 char(3),
a2 char(5),
a3 int,
constraint pk_A primary key (a1,a2),
constraint fk_A_C foreign key (a2) references C (c1)) engine=innodb;
```

3. EXERCICI DEMANATS

Siga el següent disseny lògic d'una BD:

EMPLEAT(DNI, Nom, Ap1, Direccion, data_nac, categoria, data_entrada, salari)

CP:{DNI}

VNN:{Nom, Ap1, Direccion, data_nac, data_entrada}

LÍNIA_DEMANAT(Cod, Num, quantitat, preu)

CP:{Cod, Num}

Caj:{Cod}-->PRODUCTE

Caj:{Num}-->COMANDA

VNN:{ quantitat, preu}

CLIENT (CIF, Nom, Direccion, tel, id_empleat)

CP:{CIF}

VNN:{Nom, Direccion}

Caj:{id_empleat}-->EMPRAT

PRODUCTE(Cod, descripció, preu)

CP:{Cod}

COMANDA(Num, data, client)

CP:{Num}

VNN:{data, client}

Caj:{client}-->CLIENT

Fes un SCRIPT per a la creació de les taules. Has de tindre en compte el següent:

- Les taules s'han de crear en l'ordre adequat (no es pot fer referència a una taula que no existeix encara)
- En esborrar una comanda s'han d'esborrar totes les files relacionades amb el mateix de la taula Línia_demanat.
- Si un empleat és esborrat, les files de la taula Client relacionades amb el mateix tindran la referència a la taula emprat a NULL.
- Quan es modifiquen les claus primàries de qualsevol taula els seus els camps que corresponguen de les taules que la referencien també es modificaran.
- Un producte no es pot esborrar si està referenciat en una fila de Línia_Demanat
- Una fila de Clients no es pot esborrar si ha fet comandes.
- En la taula DEMANAT Num ha d'autoincrementar-se
- En LÍNIA_DEMANAT la quantitat serà per defecte 1 i no puga ser menor que 1 (encara que el check no funcione en MySql pots incloure-ho)
- La Data de la taula Demanats serà per defecte la del sistema.

MODEL FÍSIC. EXERCICIS DDL Butlletí B

```
CREATE SCHEMA ExerDemanats;

USE ExerDemanats;

CREATE TABLE EMPLEAT (
DNI      CHAR(9) PRIMARY KEY,
nom VARCHAR(50) NOT NULL,
apl VARCHAR(50) NOT NULL,
Direccio VARCHAR(50) NOT NULL,
data_naix date not null,
categoria varchar(30),
data_entrada date not null,
Salari decimal(6,2)
) engine=innodb;

CREATE TABLE CLIENTS(
CIF CHAR(9) PRIMARY KEY,
Nom VARCHAR(50) NOT NULL,
Direccio VARCHAR(50) NOT NULL,
id_empleat CHAR(9),
CONSTRAINT FK_Clients_Empleat FOREIGN KEY(id_empleat) REFERENCES EMPLEAT(DNI) ON
DELETE SET NULL ON UPDATE CASCADE
) engine=innodb;

CREATE TABLE PRODUCTE(
cod char(5) PRIMARY KEY,
descripcio VARCHAR(100),
preu DECIMAL(5,2)
) engine=innodb;

CREATE TABLE COMANDA(
Num integer auto_increment primary key,
Data_comanda timestamp NOT NULL default CURRENT_TIMESTAMP,
clien CHAR(9) NOT NULL,
CONSTRAINT FK_Comanda_Clients FOREIGN KEY (clien) REFERENCES CLIENTS(CIF) ON
UPDATE CASCADE
) engine=innodb;

/*No es pot ficar la paraula reservada client por això he usat clien*/

CREATE TABLE LINEA_DEMANAT(
Cod CHAR(5) REFERENCES PRODUCTE(Cod),
Num integer,
quantitat integer not null,
preu decimal(5,2) not null,
constraint pk_Linea PRIMARY KEY (Cod, Num),
Constraint fk_Linea_Producto FOREIGN KEY (Cod) REFERENCES PRODUCTE(Cod) ON
UPDATE CASCADE,
Constraint fk_Linea_Comanda FOREIGN KEY (Num) REFERENCES
COMANDA(Num) ON DELETE CASCADE ON UPDATE CASCADE
) engine=innodb;
```

4. EXERCICI PRÉSTECES

Siga el següent script:

```
CREATE TABLE socis
(soci_no INT(4),
cognoms VARCHAR(14),
telefono CHAR(9) NOT NULL,
data_alta DONA'T DEFAULT '2012-01-01',
```

```

direccion VARCHAR(20),
codi_postal INT(5) CHECK (codi_postal BETWEEN 28000 AND 28999),
CONSTRAINT PK_SOCIS PRIMARY KEY(soci_no),
CONSTRAINT UQ_UNIQUE UNIQUE(cognoms),
) engine=innodb;

```

```

CREATE TABLE PRESTEM
(num_prestem INT(2) PRIMARY KEY,
soci_no INT(4),
CONSTRAINT FK_SOCIS_PRESTEM FOREIGN KEY (soci_no)
REFERENCES socis(soci_no) ON UPDATE CASCADE) engine=innodb;

```

- Amb l'esquema de dalt, que passaria hi haguera una fila de **PRESTEM** que referenciara a una altra en **SOCIS** i volguérem esborrar aquesta última fila de la taula socis? Raona la teua resposta,

No ens deixaria ja que no hi ha en la taula PRESTEM cap directriu d'esborrat (ON DELETE) i per defecte la directriu és NO ACTION

- Introdueix una columna en la taula **PRESTEM** anomenada data_prestem el valor de la qual per defecte la data del sistema (pàgines 595 i 596 del manual de MySql)

```
alter table PRESTEM add data_prestem TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
```

- Modifica la taula **SOCIS** per a introduir un camp quota que serà un real major que 5.

```
alter table socios add quota decimal(4,2) check (quota>5);
```

5. EXERCICI ESCALA

<p>Associació (codi, nom, ubicació)</p> <p>CP:{codi}</p>	<p>Pics (nom, altura, coordenades, país)</p> <p>CP: {nom}</p>
<p>Alpinistes (nif, nom, data_naixement, codi, data_ingrés)</p> <p>CP:{nif}</p> <p>CA:{codi} → Associació</p>	<p>Escala (nif, nom, data_inici, temps, oxigen, cara)</p> <p>CP{nif, nom}</p> <p>CA{nif} → alpinista</p> <p>CA{nomene} → pic</p>
<p>Telèfons (nif, telefono)</p> <p>CP:{nif,telefono}</p> <p>CA:{nif} → Alpinistes</p>	

- Tradueix el **Model Lògic Relacional** a **Model Físic en SQL**

MODEL FÍSIC. EXERCICIS DDL Butlletí B

- Crea les taules incloent totes les sentències en un arxiu. Crida al script **alpinista.sql** .

```
CREATE TABLE ASSOCIACIO(  
codi VARCHAR(4),  
nom VARCHAR(30),  
ubicacio VARCHAR(30),  
CONSTRAINT associacio_pk PRIMARY KEY (codi)) engine=innodb;  
  
CREATE TABLE ALPINISTES (  
nif CHAR(9),  
nom VARCHAR(30) ,  
data_naix DATE ,  
codi VARCHAR(4) ,  
data_ingres DATE ,  
CONSTRAINT alpinistas_pk PRIMARY KEY (nif),  
CONSTRAINT codi_ASSOCIACIO_fk FOREIGN KEY (codi) REFERENCES ASSOCIACIO(codi))  
engine=innodb;  
  
CREATE TABLE TELEFONS (  
nif CHAR(9),  
telefono CHAR(9),  
CONSTRAINT telefons_pk PRIMARY KEY (nif, telefono),  
CONSTRAINT nif_tele_fk FOREIGN KEY (nif) REFERENCES ALPINISTES(nif))  
engine=innodb;  
  
CREATE TABLE PICS (  
Nom VARCHAR(30),  
altura INTEGER ,  
coordenades VARCHAR(10) ,  
pais VARCHAR(20) ,  
CONSTRAINT pics_pk PRIMARY KEY (nombre)) engine=innodb;  
  
CREATE TABLE ESCALA (  
nif CHAR(9) ,  
nom VARCHAR(30),  
data_inici DATE ,  
temps VARCHAR(10) ,  
oxigen DECIMAL(3,2),  
cara VARCHAR(20),  
PRIMARY KEY (nif, nom),  
FOREIGN KEY (nif) REFERENCES ALPINISTES(nif),  
FOREIGN KEY (nom) REFERENCES PICS (nom)
```

MODEL FÍSIC. EXERCICIS DDL Butlletí B

```
) engine=innodb;
```

- Afegir un camp **observacions** sobre la taula **Pics** (decideix tu el tipus i grandària del mateix).

```
ALTER TABLE PICS ADD observacions VARCHAR(100);
```

- Modifica el camp **codi** de la taula **Associacio** perquè pugui contingre números fins al 999.999.

- **OPCIÓ A: (7 caràcters)**

```
ALTER TABLE ASSOCIACIO MODIFY codi varchar(7);  
ALTER TABLE ALPINISTES MODIFY codi varchar(7);
```

- **OPCIÓN B: Canviem a MEDIUMINT**

En aquest cas hauríem de donar de baixa la FK, canviar el tipus de dades de totes dues taules i tornar a crear la CONSTRAINT. Si les taules tingueren dades no-numèriques, podria donar-se un problema de conversió que ens impediria el canvi de tipus

```
ALTER TABLE ALPINISTES DROP CONSTRAINT codi_ASSOCIACIO_fk;  
ALTER TABLE ASSOCIACIO MODIFY codi MEDIUMINT;  
ALTER TABLE ALPINISTES MODIFY codi MEDIUMINT;  
ALTER TABLE ALPINISTES ADD CONSTRAINT codi_ASSOCIACIO_fk FOREIGN KEY (codi)  
REFERENCES associacio(codi);
```

- Afegir un camp **foto** sobre la taula ALPINISTES (decideix tu el tipus i grandària del mateix).

```
ALTER TABLE ALPINISTES ADD (foto BLOB);
```

- Modifica la columna **data_ingrés** de la taula ALPINISTES perquè per defecte tinga la data del sistema.

En MySQL o MariaDB el tipus DATE no es pot actualitzar automàticament, només el fa el tipus TIMESTAMP a CURRENT_TIMESTAMP. Per tant, si volem especificar una data per defecte, canviarem primer el tipus del camp de DATA a TIMESTAMP.

```
ALTER TABLE ALPINISTES MODIFY fecha_ingreso timestamp DEFAULT CURRENT_TIMESTAMP;
```

- El valor del camp '**coordenades**' no ha de repetir-se mai, malgrat no ser clau primària. Introdueix la corresponent restricció sobre **Pics**.

Dues opcions:

MODEL FÍSIC. EXERCICIS DDL Butlletí B

```
ALTER TABLE PICS add unique (coordenades);
```

```
ALTER TABLE PICS ADD CONSTRAINT pic_coo_uk UNIQUE (coordenades);
```

- Si s'esborra un alpiniste de la base de dades què hauria de fer-se amb les seues corresponents entrades en la taula **Telèfons**? Indica la sentència DDL necessària per a aplicar aquesta restricció.

```
/*Primer borrem la clau aliena*/  
ALTER TABLE telefons DROP CONSTRAINT nif_tele_fk;  
  
/*tornem a crear-la amb la directriu de borrat*/  
ALTER TABLE telefons ADD CONSTRAINT nif_tele_fk FOREIGN KEY (nif)  
REFERENCES alpinistes(nif) ON DELETE CASCADE;
```

- I si s'esborra una associació de la base de dades?

```
ALTER TABLE ALPINISTES DROP CONSTRAINT codi_ASSOCIACIO_fk;  
ALTER TABLE ALPINISTES ADD CONSTRAINT codi_ASSOCIACIO_fk FOREIGN KEY (codi)  
REFERENCES associacions(codi) ON DELETE SET NULL;
```

6. EXERCICI SUCURSALS

Tenim les següents taules creades, però no s'ha creat cap clau aliena. Es tracta que vosaltres analitzeu quines són les claus alienes necessàries per al correcte funcionament del sistema i les afegiu a la base de dades per mitjà de modificacions de les taules

```
CREATE TABLE ARTICLES(  
  id_art INTEGER,  
  nom VARCHAR(30) default '',  
  preu NUMBER default 0.0,  
  codi VARCHAR(7) default '',  
  id_gru INTEGER default 0 NOT NULL,  
  CONSTRAINT ART_ID_PK PRIMARY KEY(id_art)) engine=innodb;  
  
CREATE TABLE CLIENTES (  
  id_cli INTEGER NOT NULL ,  
  nom VARCHAR(60) default '' NOT NULL ,  
  direccio VARCHAR(80) default '',  
  constraint cli_idc_pk PRIMARY KEY (id_cli)) engine=innodb;
```


MODEL FÍSIC. EXERCICIS DDL Butlletí B

```
CREATE TABLE FACTURES (  
  id_fac INTEGER NOT NULL ,  
  serie VARCHAR(1) default '' NOT NULL,  
  num INTEGER default 0 NOT NULL,  
  any date,  
  id_cli INTEGER default 0 NOT NULL,  
  id_vene INTEGER NOT NULL,  
  CONSTRAINT FAC_IDF_PK PRIMARY KEY (id_fac)) engine=innodb ;  
  
CREATE TABLE GRUPS (  
  id_gru INTEGER,  
  descripcio VARCHAR(15) default '' NOT NULL,  
  CONSTRAINT GRU_IDG_PK PRIMARY KEY (id_gru)) engine=innodb;  
  
CREATE TABLE LINIES_FACTURA (  
  id_lin INTEGER NOT NULL,  
  id_fac INTEGER default 0 NOT NULL ,  
  impo FLOAT default 0 NOT NULL ,  
  id_art INTEGER NOT NULL,  
  quantitat INTEGER NOT NULL,  
  any data default '01/01/2000' NOT NULL ,  
  id_suc INTEGER default 0 NOT NULL ,  
  constraint LIN_PK PRIMARY KEY (id_lin, factura)) engine=innodb;  
  
CREATE TABLE SUCURSALS (  
  id_suc INTEGER NOT NULL,  
  descripcio VARCHAR(15) default '' NOT NULL ,  
  CONSTRAINT SUC_IDS_PK PRIMARY KEY (id_suc)) engine=innodb;  
  
CREATE TABLE VENEDORS (  
  id_vene INTEGER NOT NULL,  
  nom VARCHAR(50) default '' NOT NULL ,  
  data_ingres date default '01-01-2000' NOT NULL ,  
  salari float default 0 NOT NULL ,  
  CONSTRAINT VENE_IDV_PK PRIMARY KEY (id_vene)  
  ) engine=innodb;
```

MODEL FÍSIC. EXERCICIS DDL Butlletí B

SOLUCIÓ:

```
ALTER TABLE ARTICLES
  ADD CONSTRAINT art_idg_fk FOREIGN KEY (id_gru) REFERENCES GRUPS (id_gru)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE FACTURES
  ADD CONSTRAINT fac_idv_fk FOREIGN KEY (id_vene) REFERENCES VENEDORS (id_vene)
  ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT fac_idc_fk FOREIGN KEY (id_cli) REFERENCES CLIENTES (id_cli)
  ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE lineas_factura
  ADD CONSTRAINT linf_ids_fk FOREIGN KEY (id_suc) REFERENCES SUCURSALS (id_suc)
  ON UPDATE CASCADE,
  ADD CONSTRAINT linf_idf_fk FOREIGN KEY (id_fac) REFERENCES FACTURES (id_fac)
  ON UPDATE CASCADE,
  ADD CONSTRAINT linf_ida_fk FOREIGN KEY (id_art) REFERENCES ARTICLES (id_art);
```